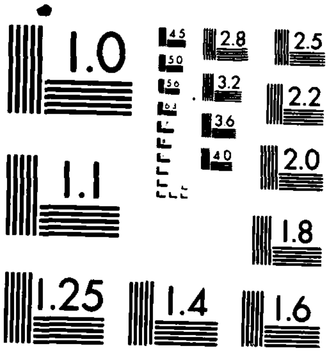
AD-A161 875 TEXT-DEPENDENT SPEAKER VERIFICATION USING VECTOR QUANTIZATION SOURCE CODING(U) NAVAL RESEARCH LAB WASHINGTON DC D K BURTON 2 NOV 85 NRL-MR-5662 1/1

UNCLASSIFIED F/G 17/2 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

1

NRL Memorandum Report 5662

Text-Dependent Speaker Verification Using Vector Quantization Source Coding

D. K. BURTON

Computer Science and Systems Branch
Information Technology Division

November 26, 1985

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION / DOWNGRADING SCHEDULE

3 DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4 PERFORMING ORGANIZATION REPORT NUMBER(S)
NRL Memorandum Report 5662

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Naval Research Laboratory

6b OFFICE SYMBOL *(If applicable)*
Code 7591

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS *(City, State, and ZIP Code)*
Washington, DC 20375-5000

7b. ADDRESS *(City, State, and ZIP Code)*

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Office of Naval Research

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS *(City, State, and ZIP Code)*
Arlington, VA 22217

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
61153N		RR014-09-41	DN980-167

11 TITLE *(Include Security Classification)*
Text-Dependent Speaker Verification Using Vector Quantization Source Coding

12 PERSONAL AUTHOR(S)
Burton, D.K.

13a. TYPE OF REPORT
Interim

13b. TIME COVERED
FROM 10/84 TO 6/85

14. DATE OF REPORT *(Year, Month, Day)*
1985 November 26

15. PAGE COUNT
22

16 SUPPLEMENTARY NOTATION

17 COSATI CODES

FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Speaker verification, Vector quantization
Speaker recognition, Matrix quantization

19 ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Several vector quantization approaches to the problem of text-dependent speaker verification are described. In each of these approaches, a source codebook is designed to represent a particular speaker saying a particular utterance. Later, this same utterance is spoken by a speaker to be verified and is encoded in the source codebook representing the speaker whose identity was claimed. The speaker is accepted if the verification utterance's quantization distortion is less than a prespecified speaker-specific threshold. The best of the approaches achieved a 0.7% false acceptance rate and a 0.6% false rejection rate on a speaker population containing 16 admissible speakers and 111 casual imposters. The approaches are described, and detailed experimental results are presented and discussed.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT □ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a NAME OF RESPONSIBLE INDIVIDUAL
David K. Burton

22b TELEPHONE *(Include Area Code)*
(202) 767-3490

22c. OFFICE SYMBOL
Code 7591

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

CONTENTS

Text-Dependent Speaker Verification Using Vector Quantization Source Coding

I. INTRODUCTION

Speaker verification by machine consists of automatically authenticating the identity claimed by a speaker given only samples of the speaker's voice. It has been an area of active study for more than twenty years, and two categories of approaches to this problem have developed. In one, verification decisions are based on speech that is selected by the speaker and not known ahead of time by the verification system. This is known as text-independent verification. In the other category, the verification system is trained on a prespecified utterance and later this same utterance is spoken by the individual in question – this is called text-dependent speaker verification. In this paper, we describe and evaluate several new approaches to the text-dependent speaker verification problem.

A typical approach to text-dependent speaker verification consists of the following. Select a parameter or a set of parameters that can be derived from the speech waveform and then represent each speaker by a time-series of these parameters (called a reference template) obtained from a particular utterance. The parameters are chosen normally with the hope that they reflect speaker-specific, organic differences in the structure of the vocal apparatus or, perhaps, that the time series of parameters will reflect learned differences in the use of the vocal apparatus to produce a particular utterance. After obtaining a reference for each speaker, an unknown speaker claims an identity and speaks an appropriate utterance. This utterance is analyzed and a time-series of parameters is obtained. The unknown speaker's parameters are aligned in time with the reference stored for the speaker whose identity was claimed, and the decision to accept or reject the speaker is based on a measure of the similarity between the two time-series of parameters. Examples of parameters that have been used in this way are pitch [1], short time energy [1], short time spectra [2], and linear predictive coding (LPC) coefficients or parameters that can be derived from these coefficients [3].

In addition to the template matching approach described above, statistical methods have been studied [4, 5]. These methods use large amounts of training data to estimate the underlying probability densities for the parameters chosen to represent a speaker. Once the probability densities are specified, statistical detection theory methods are used to verify a speaker [6].

We approach the text-dependent speaker verification problem from a different viewpoint. We consider a speaker of a particular utterance as an information source, and we model this information source using a standard information-theoretic source coding method called vector quantization (VQ). VQ is a source coding technique [7] that has been used successfully in both speech coding [8] and speech recognition [9, 10, 11]. In VQ, each source vector is coded as one of a prestored *set of codewords*, called a *codebook*, by finding the codeword that minimizes the distortion between itself and the source vector. For speech, a codebook is designed from a training sequence consisting of typical speech [12]. The training sequence is divided into frames (typically 20 milliseconds), linear predictive analysis is done on each frame, and a clustering algorithm is used on this sequence of LPC coefficients to obtain a codebook of representative spectra, or codewords. The codebook is designed to minimize the average quantization distortion between itself and the training sequence.

Manuscript approved July 16, 1985.

To use VQ source coding in speaker verification, we represent each speaker by a VQ codebook designed from a training sequence composed of repetitions of a particular utterance. Later, this same utterance is spoken by an unknown speaker with a claimed identity. This test utterance is coded in the codebook representing the speaker whose identity was claimed, and the resulting quantization distortion is compared to a threshold. If the distortion is less than the threshold, the speaker is accepted.

In addition to our source coding point of view, our speaker verification approach is quite different from other approaches in several ways. No attempt is made to align-in-time a test sequence with a stored reference sequence (indeed, no reference sequence exists), and no explicit estimate is made of an underlying probability density function. Our verification procedures are, however, closely related to optimal information-theoretic methods of classification that use the ***information dissimilarity*** between two vectors as a discrimination measure [13, 14]. Preliminary results were reported in [15].

We previously used VQ source coding in isolated word recognition [9, 16, 17]. The methods used in those approaches to represent a word (design a codebook) and to compare an unknown input word with the stored codebooks (classify an input utterance) are the same as the ones described in this paper to represent and verify a speaker. The differences are in the application of the ideas and in the use of thresholds to make decisions.

Recently, another VQ based approach to speaker identification has been reported by Soong ***et al*** [18]. In this approach, each speaker is represented by a single VQ codebook designed to represent that speaker saying the 10 digits. An unknown speaker then says the 10 digits, and the average quantization distortion resulting from encoding the digits is used as a discrimination measure to identify the speaker. Reported results are quite good – an error rate less than 2% [18].

The rest of this paper is organized as follows. Section II describes three ways to represent a speaker by using VQ source coding. Section III explains our speaker verification approach. Section IV presents experimental results, and section V concludes with a summary and general discussion.

II. BACKGROUND

In this section we briefly describe three ways to design a source model of a speaker; for detailed descriptions of the methods, see [9, 16, 17]. Following these descriptions is a list of the distortion measures and LPC parameters we used in the speaker verification experiments.

First we establish some notation and define some terms. Upper and lower case roman and italic letters (e.g. n, N, q, Q) denote scalars; lower case italic letters with bars (e.g. $\bar{c}$) denote vectors; upper case italic letters with bars (e.g. $\bar{C}$) denote sets of vectors (e.g. $\bar{C} = \{\bar{c}_1, \bar{c}_2, \cdots, \bar{c}_N\}$); bold lower case roman letters (e.g. $\mathbf{c}$) denote sequences of vectors (e.g. $\mathbf{c} = \bar{c}_i\,;\ i=1, \cdots, K$); and bold upper case roman letters (e.g. $\mathbf{C}$) denote sets of vector sequences (e.g. $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_N\}$).

Throughout, all vectors consist of LPC coefficients and a gain term. $\bar{T} = \{\bar{t}_1, \bar{t}_2, \cdots, \bar{t}_P\}$ is a P-vector training sequence obtained from M repetitions of an utterance by a speaker. $\bar{V} = \{\bar{v}_1, \bar{v}_2, \cdots, \bar{v}_L\}$ is an L-vector test sequence corresponding to an utterance obtained from a speaker for verification purposes. $\bar{C}$ represents a VQ codebook, whether it is single-section or multisection will be clear from the context, and finally, $\mathbf{C}$ represents a matrix quantization codebook.

A. Single Section Vector Quantization

For speaker verification, a single-section VQ codebook $\bar{C}$ is designed to minimize the average distortion that results from encoding a training sequence $\bar{T}$

$$\sum_{p=1}^{P} d(\overline{t}_p, \overline{c}_B), \tag{1}$$

where $\overline{c}_B$ is the codeword resulting from encoding the speech segment $\overline{t}_p$,

$$d(\overline{t}_p, \overline{c}_B) = \min_i d(\overline{t}_p, \overline{c}_i),$$

and d is an appropriate vector distortion measure. This codebook represents a speaker saying a particular word.

The average quantization distortion D_{avg} that results from coding a verification utterance $\overline{V}$ in codebook $\overline{C}$ is

$$D_{avg} = \frac{1}{L}\sum_{l=1}^{L} d(\overline{v}_l, \overline{c}_B). \tag{2}$$

It is this average quantization distortion that is used in making the verification decision.

This approach is called single section to distinguish it from the approach described in the next section in which each speaker is represented by a codebook consisting of a sequence of single-section codebooks.

B. Multisection Vector Quantization

In multisection VQ, we represent each speaker by a time-dependent sequence of single section codebooks, which we call a multisection codebook. A speaker is verified by dividing his verification utterance $\overline{V}$ into sections that correspond to the sections of the multisection codebooks, doing VQ on a section-by-section basis with the appropriate multisection codebook, and computing the average distortion.

To be more specific, let F_q be the number of frames in the q^{th} utterance in the training sequence for $\overline{C}$, where $q=1, \cdots, M$; and let U_{mq} be the m^{th} frame in the q^{th} training utterance where $m=1,...,F_q$. Now the multisection codebook $\overline{C}$ consists of a sequence of VQ *section codebooks* $\overline{C}_j$, where the section codebook $\overline{C}_j$ is designed using (1) and n frames from each training utterance. That is, $\overline{C}_j$ is designed from the frames U_{mq}, where $m=(j-1)n+1,...,jn$, and $q=1,...,M$. For example, $\overline{C}_1$ is designed from the first n frames of each training utterance, $\overline{C}_2$ from the second n frames, etc. We call n the *section length* – it is the number of frames that are spanned per section. Finally, let $\overline{c}_{ji}$, $i=1, \ldots, N_j$ be codewords in section codebook $\overline{C}_j$.

D_{avg} is the average distortion resulting from coding the verification utterance $\overline{V}$ with the codebook $\overline{C}$,

$$D_{avg} = \frac{1}{L}\sum_{j=1}^{S} d_j, \tag{3}$$

where S is the number of section codebooks in $\overline{C}$,

$$d_j = \sum_{l=(j-1)n+1}^{\min[jn,\, L]} \min_i d(\overline{v}_l, \overline{c}_{ji}),$$

is the total distortion from coding the j^{th} section of the utterance $\overline{V}$ with the j^{th} section codebook $\overline{C}_j$ of $\overline{C}$, and n is the section length. The verification decision is made using this distortion.

C. Matrix Quantization

In matrix quantization, instead of coding a *single* source vector in a codebook containing characteristic vectors, we code a *time-ordered sequence* of source vectors in a codebook containing characteristic vector sequences. Given $\overline{T}$, we find the matrix quantization codebook $\mathbf{C}$ containing codeword matrices $\mathbf{c}_j = [\overline{c}_{j1}, \overline{c}_{j2}, ..., \overline{c}_{jK}]$ that minimizes

$$\sum_{p=1}^{P-K+1} D(\mathbf{t}_p, \mathbf{c}_B),$$

where $\mathbf{c}_B$ is the codeword matrix resulting from coding the sequence of training vectors

$$\mathbf{t}_p = [\bar{t}_p, \bar{t}_{p+1}, \ldots, \bar{t}_{p+K-1}],$$

by using the nearest neighbor rule

$$D(\mathbf{t}, \mathbf{c}_B) = \min_j D(\mathbf{t}, \mathbf{c}_j),$$

and where the distortion between a speech segment $\mathbf{t}$ and the j^{th} codeword is

$$D(\mathbf{t}, \mathbf{c}_j) = \sum_{l=1}^{K} d(\bar{t}_l, \bar{c}_{jl}). \tag{4}$$

We call K the *codeword matrix size*. The MQ codebook design algorithm we used [19] is a generalized version of the VQ design algorithm developed by Linde *et al* [12].

To use MQ in speaker verification, we represent a speaker saying a particular word by a codebook $\mathbf{C}$, just as in the VQ approaches above. A verification utterance is processed by dividing it into overlapping sequences of K frames, coding each K frame sequence in the speaker-codebook $\mathbf{C}$, and computing the average quantization distortion between the utterance and the codebook. To be specific, for a verification utterance $\overline{V}$, the average distortion resulting from coding it with codebook $\mathbf{C}$ is

$$D_{avg} = \frac{1}{L} \sum_{l=1}^{L-K+1} D(\mathbf{v}_l, \mathbf{c}_B). \tag{5}$$

D. Distortion Measures

Based on results from previous work on isolated word recognition [9], we used the *gain normalized Itakura-Saito* distortion measure (d_{GN}) in (1) and (4) to generate codebooks. For power spectrum estimates f and f that have the autoregressive (LPC) form

$$f(\theta) = \frac{\sigma^2}{|A(z)|^2},$$

where

$$A(z) = \sum_{k=0}^{M} a_k z^{-k}$$

and $z = \exp(i\theta)$, the d_{GN} distortion is given by

$$d_{GN}(f, \hat{f}) = \frac{\alpha}{\sigma^2} - 1,$$

where

$$\alpha = r(0)\hat{r}_a(0) + 2\sum_{n=1}^{M} r(n)\hat{r}_a(n),$$

$$\hat{r}_a(n) = \sum_{i=0}^{M-n} \hat{a}_i \hat{a}_{i+n},$$

and where $r(n)$ are the time-domain autocorrelations of $f(\theta)$. For the verification distortion measure in (2), (3), and (5), we used the *gain optimized Itakura-Saito* distortion measure (d_{GO}),

$$d_{GO}(f, \hat{f}) = \ln(\alpha) - \ln(\sigma^2),$$

which is also known as the log likelihood distortion measure. Properties of these distortion measures are discussed in [20].

E. LPC Parameters

LPC parameters for both codebook generation and speaker verification were generated using the autocorrelation method of linear predictive analysis with Hamming windowing. We chose analysis conditions for compatibility with the Navy's 2.4-kbs LPC-10 system[21]: analysis window width = 128 points, filter order = 10, and pre-emphasis = 94%.

III. SPEAKER VERIFICATION APPROACH

Usually, no information is available for the characteristics of specific unacceptable speakers, and the main problem in applying these source coding approaches to speaker verification is to formulate a criterion for rejecting a speaker. To decide whether to reject a speaker (given an utterance), we associate a threshold with each speaker codebook. An unknown utterance (speaker) is rejected if its distortion exceeds the threshold. To design thresholds for a speaker, we estimate parameters for two Gaussian distributions: the *in-class* distribution of distortions (obtained by encoding utterances from that speaker in his or her codebook) and the *out-of-class* distribution of distortions resulting from encoding utterances spoken by other speakers. We choose the threshold to equalize the overlap area of the two distributions, thus equalizing the expected numbers of imposter acceptances (false acceptances) and rejections of acceptable speakers (false rejections).

In more detail, the threshold computation is as follows. For each speaker, encode that speaker's training data with his or her codebook. Compute the mean distortion μ_i^{in} resulting from encoding the training data from speaker i in speaker i's codebook, and compute the corresponding standard deviation σ_i^{in}. Also compute μ_i^{out}, the mean distortion resulting from encoding utterances *not* spoken by speaker i using the codebook for speaker i, and the corresponding standard deviation σ_i^{out}. To equalize the number of false acceptances and false rejections, the threshold T_i is chosen to be an equal number of standard deviations away from each mean, giving

$$T_i = \frac{\mu_i^{\mathrm{in}}\sigma_i^{\mathrm{out}} + \mu_i^{\mathrm{out}}\sigma_i^{\mathrm{in}}}{\sigma_i^{\mathrm{out}} + \sigma_i^{\mathrm{in}}}.$$

This method of threshold determination assumes Gaussian distributions. Some previous studies by Buck [22], however, showed that the logarithms of average distortions are more nearly Gaussian than the distortions themselves; so the thresholds were based on the statistics of the logarithms of distortions, instead of simply the distortion as shown in (2), (3), and (5).

To verify a speaker, the verification utterance $\overline{V}$ is coded in the appropriate codebook and the average log distortion is computed. This distortion value is compared to the threshold associated with that codebook, and if the distortion value exceeds the threshold, the speaker is rejected; otherwise the speaker is accepted.

Preliminary experiments indicated that verification accuracy using a single verification utterance is poor [15]. To improve the verification accuracy, we based the verification decision on the results for several words. The next section describes our approach to extending this method to multiple words.

A. Extension To Multiple Words

In previous work [15], we examined three ways of extending our method to more than one word. All three methods achieved about the same verification accuracy, and based on those results, we used the simplest of the three methods in this work.

For each speaker, a separate codebook is designed for each word; if W words are to be spoken, there are W codebooks for each speaker. Separate thresholds are computed for each word, and W different verification decisions are made. For example, if a speaker is requested to say *zero* , *three* , and *nine* , the *zero* utterance is encoded with the *zero* codebook from that speaker; the *three* utterance is encoded with the *three* codebook; etc. To make a verification decision, we use a majority rule; the decision made by a majority of the individual word classifiers is used as the overall decision. In case of ties, the speaker is rejected.

IV. EXPERIMENTS

We first describe the speech data bases that were used in the verification experiments. We next describe how the data bases were partitioned for use in separate parameter studies and evaluation tests and what parameters were varied in the studies. This is followed by three subsections; each subsection describes the verification results using one of the source models described in section II.

A. Data Bases

We combined two data bases to do these experiments, both collected by Texas Instruments Inc. (TI). The main difference in the data bases is the resolution of the A/D converters. One data base was digitized with a 12-bit converter; the second was digitized with a 16-bit converter.

Data for designing the codebooks to represent the speakers, determining the parameters for the in-class distributions, and testing verification accuracy came from the data base described in [23]. It contains 26 utterances of each digit (*zero* through *nine*) by 16 speakers (8 male and 8 female). We call this data base TI-1. The data used for determining the parameters for the out-of-class distributions and for testing the imposter rejection capabilities of the methods came from a data base designed for evaluating speaker-independent recognition of the digits [24]. It contains two utterances of each of the 10 digits from 109 adult male and 111 adult female speakers that are distinct from the speakers in TI-1. This data base is divided into two parts: a *training* part containing 54 male and 55 female speakers, and a *testing* part containing 54 male and 57 female speakers. We call this second data base TI-2.

Automatic endpoint detection for both training and test utterances was used in our experiments. Our endpoint-detection algorithm is based on ideas presented in [25, 26]. Briefly, the algorithm first analyzes the background noise to determine its average magnitude and then uses the result to set various thresholds that are used to find significant "energy clumps" in the data. See [9] for more details.

B. Data Base Partition

We first determined the number of training utterances required to characterize a speaker saying a digit. To do this, for each speaker–digit combination, we designed a series of 8-codeword, single section codebooks. We designed the first codebook using a one-utterance training sequence, and increased the number of training utterances by one for each new codebook. We recorded the average codebook-design distortion for each codebook, and after designing all the codebooks, examined the results looking for the number of training utterances required to maximize the codebook-design distortion. (See Figure 1 for the results from a typical speaker.) On average, it took 8 utterances to reach 90% of the maximum codebook-design distortion, and based on this, we designed codebooks using 8 training utterances in all our experiments.

In all three parameter studies described below, we designed digit codebooks for each speaker in the TI-1 data base from the first 8 utterances of each digit. These 8 training utterances plus the next 4 utterances were used to estimate the parameters for the in-class distribution for each speaker–word model. The next 7 digits in TI-1 were the speaker supplied verification data. For the parameter studies, the TI-2 training data was divided into two parts. One part, containing the first 27 male and 27 female speakers, was used to estimate the out-of-class distribution parameters; the second part, containing the rest of the training portion of TI-2, was used as imposter data.

Based on the results of the parameter studies, we chose several sets of words and codebook parameters to use in the full data base tests of the three source models. In these tests, the training data for each speaker codebook again consisted of the first 8 utterances of a digit. The in-class parameter estimation data, however, consisted of the first 16 utterances. The remaining 10 utterances of each digit were the verification data. We used all 109 speakers in the training portion of TI-2 to estimate the out-of-class distribution parameters and the 111 speakers in the test portion as the imposters.

C. Experimental Parameters

The codebook size, or the number of codewords in a codebook, is a parameter that we varied in the experiments. For single section codebooks, the codebook size is always a power of 2 – i.e., $N = 2^R$, and we call R the rate of the codebook. For multisection codebooks, the size of the constituent section codebooks is also always a power of 2, and we call the section codebook rate R_S. Similarly, the size of matrix quantization codebooks is a power of 2; we call the matrix codebook rate R_M.

In addition to the codebook rates, we varied the section length n for multisection codebooks and the matrix size K for matrix quantization codebooks during the parameter studies. The parameters used during verification always matched those used in designing the codebooks.

There are a number of factors affecting the design of codebooks and thus the verification results that we did not vary. For one, we preprocessed the training and verification data by dividing each utterance into 24 equal length frames. This was done to provide a rough form of normalization. Also, for single-section and multisection codebooks, we used an energy (sum-of-squares of data points) threshold of 250 to ignore low energy frames; this threshold was used both in codebook generation and speaker verification. For matrix quantization, we handled low energy frames in the following manner. The first $K-1$ low-energy frames in a sequence were replaced with flat-spectrum frames with energy equal to 250; if more than $K-1$ frames occurred in a sequence, we ignored all but the first $K-1$. The reason for this was to preserve transitions from silence-to-speech and vice versa, while eliminating any all-silent training and verification segments.

D. Single Section Results

Parameter Studies. We varied the codebook rate R in these experiments. Verification decisions were made using all 10 digits and the majority-rule classifier that was described in section III.A. The results are listed in Table I for R ranging from 1 to 4. Most of the verification errors were false rejections. This implies that the acceptance thresholds are too small. Because of this, in the verification tests (described in this and the next two subsections IV.E and IV.F), we increased the number of training utterances used to estimate the in-class distribution parameters. For R equal to 3 and 4, all errors were caused by just 2 speakers.

For R equal to 3, we measured the verification accuracy of each digit individually; the results are in Table II. No single digit reliably verifies the speakers, and the individual digit results are also biased toward false rejections. The last column in Table II contains the square root of the product of the false-acceptance rate and the false-rejection rate ($\sqrt{FA*FR}$); this is considered a good overall performance measure [27].

Verification Tests. As noted above, we added more utterances to the data that was used to estimate the in-class distribution parameters; the new training set contained 16 utterances. We felt that by including more utterances that were not in the codebook training set, the in-class distribution for a speaker would better represent new utterances from that speaker. Rate-3 codebooks were used in the verification tests because they did best in the parameter study and also because rate-3 codebooks yielded good speaker-trained isolated word recognition results [9].

The results, using all 10 digits in the verification decision, are listed by individual speaker in Table III. The majority of the errors were caused by KAB and GRD; these also were the two difficult speakers in the parameter study. The results are still biased toward false rejections, although the bias is smaller than it was in the parameter study. $\sqrt{FA*FR}$ for this test was 0.9.

Next several subsets of the digits were tested, each consisting of 5 digits. Based on the single digit parameter study, we used the best (01247), the worst (35689), and an arbitrary (25678) set of five digits. Results are listed in Table IV. Again KAB and GRD were difficult speakers, but now, many other speakers contributed to the errors. Based on the $\sqrt{FA*FR}$ values for these tests, the verification accuracies obtained by representing each speaker by five words were significantly worse than those obtained using all 10 words. Generally, the degradation in performance was restricted to false acceptances, and the overall performance was closer to the design goal of equal error rates for the two types of errors.

E. Multisection Results

Parameter Studies. We varied both the section length n and the section codebook rate R_S in these experiments. Table V shows the results. Generally for a fixed R_S value, better results are achieved using smaller n values. For $R_S=2$, tests using n equal to 1 and 2 were not done because of insufficient codebook training data. Using $n=4$ and $R_S=2$, we tested the verification performance of the individual digits; the results are in Table VI. Again as in the single section approach, no single digit gives good overall results and the errors are biased toward false rejections.

Verification Tests. We used $n=4$ and $R_S=2$ for the verification tests. These conditions were chosen because they did well both in the parameter study and in previous isolated word recognition work [16]. Table VII contains the results using all 10 digits to make the verification decision. No speaker was particularly difficult, as KAB and GRD were when using the single section approach, and in general, the results are closer to the design goal of equal false-rejection and false-acceptance error rates than were the single section results. $\sqrt{FA*FR}$ was 0.6.

Again, we did verification tests using the best (12467), the worst (03589), and an arbitrary (01234) set of five digits; the results are in Table VIII. The verification performance of the various five-digit subsets corresponded well with the expected performance based on the single digit study – i.e., the best five-digit set had the smallest $\sqrt{FA*FR}$, the worst set had the largest $\sqrt{FA*FR}$, and the arbitrary set had an $\sqrt{FA*FR}$ between the other two. The only consistently difficult speaker in these tests was KAB; averaged over the 3 five-digit tests, he had a false acceptance rate of 3.3%.

F. Matrix Quantization

Parameter Study. We varied the codebook rate R_M and the matrix size K in these experiments. For each K value, the maximum R_M was limited by the amount of codebook training data (poor codebooks often result if insufficient training data is used). The results are listed in Table IX. No obvious relationship between K and R_M is shown in these results. Using $R_M = 3$ and $K = 8$ (these conditions are also good for isolated word recognition[17]), we measured the verification performance of the individual digits; these results are in Table X. As in the single section and multisection apı roaches, the error rates are biased toward false rejections.

Verification Tests. The full data base results using $R_M = 3$, $K = 8$, and all 10 digits are listed in Table XI. Once again, KAB was a difficult speaker. We tested the best (12467), the worst (03589), and an arbitrary (01234) five digits; the verification results are in Table XII. The relative performance of the five-digit sets did not correspond exactly with the expected results based on the individual digit performances, but the worst digit-set did produce the poorest results.

V. SUMMARY AND DISCUSSION

The verification performance ($\sqrt{FA*FR}$) for the three source models when using only a single digit per speaker were similar – roughly varying from 4 to 8 depending on the digit, and consistently, the digits 1,2,4 and 7 individually did best in the speaker verification tests. When the individual digits were joined with the majority rule classifier, however, the verification performances of the three approaches were no longer equivalent. The multisection VQ source model did best when using the 10- and 5-digit sets of verification words (for the 10 digits, $\sqrt{FA*FR} = 0.6$; for the best 5 digit set, $\sqrt{FA*FR} = 0.7$). In addition, the multisection VQ approach came closer to satisfying the design goal of equal error rates, and the results on the 5-digit subsets corresponded more closely to the expected results (based on $\sqrt{FA*FR}$ for the individual digits). The next best source model was the single section approach, although the differences in $\sqrt{FA*FR}$ values between the single section VQ and the MQ approach were small.

All three source models did well in the speaker verification tests, and in retrospect, the similarity of their performances is not surprising. The three approaches are intimately connected through the codebook design algorithm, and both the multisection VQ and the MQ approaches are generalizations of single section VQ. This can be seen by considering the multisection VQ approach with a section length n equal to the normalization length (24 in this study), and considering the MQ approach with the matrix size K equal to 1. Each approach reduces to single section VQ under the appropriate condition.

The single section VQ source model captures only the short-time spectrum shape information. This spectrum shape information is useful in speaker verification because it contains estimates of formant frequencies, relative formant amplitudes, and formant bandwidths, and these are correlated with the locations and physical sizes of the speech articulators. As such, the single section results are a measure of how well the short-time spectrum can characterize a speaker. In addition, because the codebook spectra are unordered, the single section VQ source model is directly applicable to text-independent speaker verification [18]. It is generally believed, however, that examining parameters as a function of time is valuable in speaker verification for two reasons: (1) many of the speaker-characteristic properties of speech are the result of idiosyncrasies in the speaking habits of people and (2) by considering the time sequence of parameters, the emphasis is on how the parameters vary rather than the exact value of a parameter. The multisection VQ and the MQ approaches represent two different ways of incorporating some phonetic duration information into the verification process while maintaining the information-theoretic source model approach. Multisection VQ improves the verification performance, and at best, MQ does not degrade the performance. It is unclear why the durational information provided by the MQ approach does not improve the verification performance.

In addition to phonetic durations, a speaker will say an utterance with characteristic tones or intonations, and stresses [28, 1]. Because these are roughly independent of the spectrum shape information, improvements in the verification accuracy could probably be achieved by adding pitch and short time energy information to the verification process.

As an aside, it is interesting to consider how VQ speech coding could defeat a speaker verification or identification system. Our single section VQ results show that the source model found by using the Linde, Buzo, and Gray clustering algorithm [12] is an accurate representation of the short-time spectra produced by a speaker. To impersonate a speaker, one needs only to obtain training data spoken by that speaker and to design a VQ codebook specifically for that speaker. Anyone could talk through this codebook (via VQ speech coding), and the resulting speech would be characteristic of the speaker who provided the training data. It seems that this procedure would defeat any speaker recognition system that relies solely on short-time spectrum representations.

Finally, the connection between these speaker verification approaches and our previous isolated word recognition approaches needs to be emphasized. The parameters (codebook size, section length, and matrix size) and the source model (codebook) design procedure used in each of the speaker verification tests are exactly those used in our previous work on isolated word recognition [9, 16, 17]. In those studies, accuracies for speaker trained recognition of the digits exceeded 99%. The very good speaker verification and isolated word recognition results achieved using these approaches point toward a combined speaker–speech recognition system. These results also illustrate the power of the VQ source coding approach using the Linde, Buzo, and Gray clustering algorithm [12].

ACKNOWLEDGMENTS

I thank J. Shore for helpful discussions and comments on this paper, J. Buck for helpful discussions and for writing most of the software, and R. Johnson for pointing out the VQ speaker impersonation scheme. In addition, I thank T. Schalk and G. Leonard for their help in obtaining the data bases.

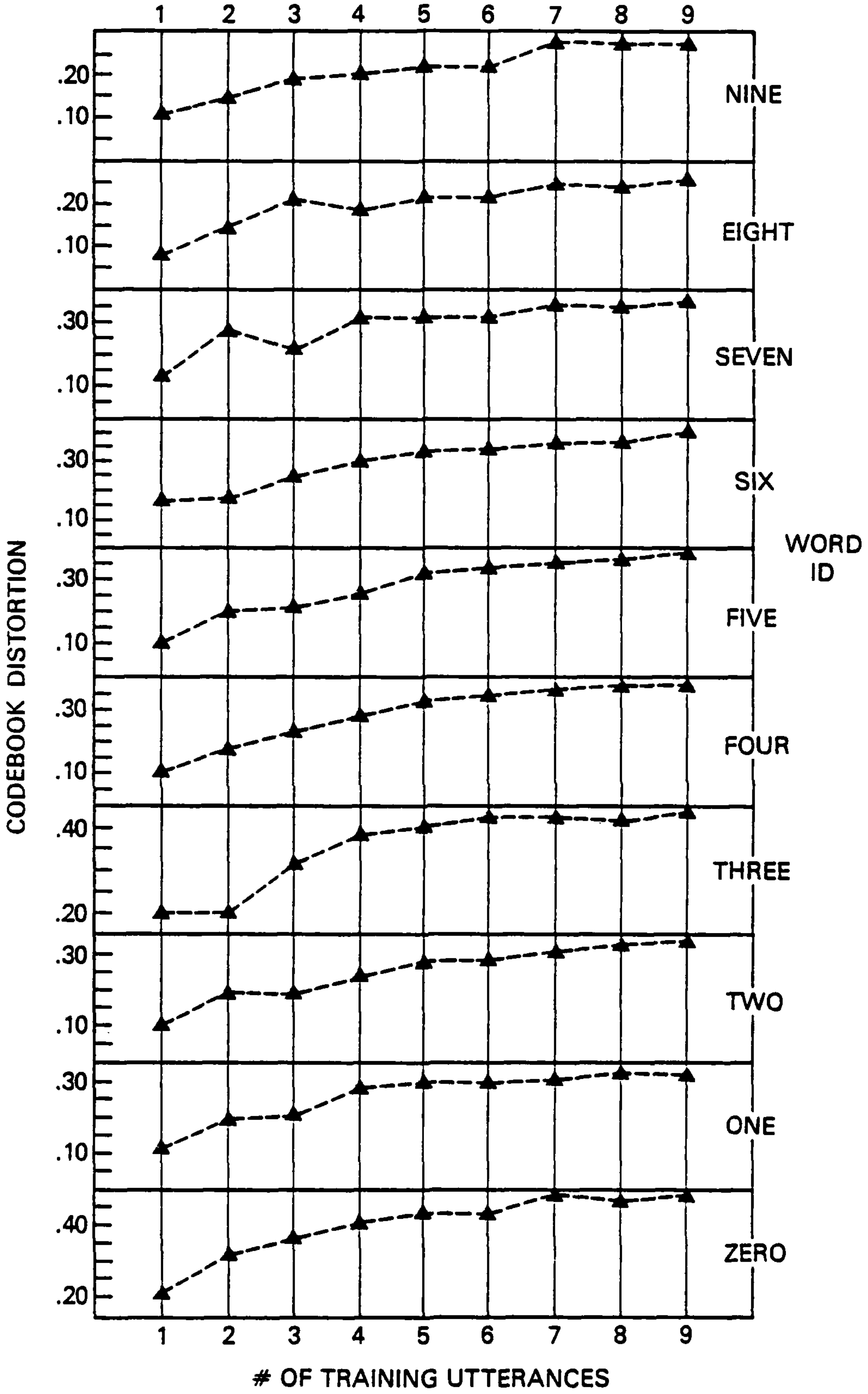


Figure 1. The codebook design distortion for each of the 10 digits as a function of the number of codebook training utterances – one speaker's results.

Table I. Speaker Verification Study: Single Section Codebooks, Majority Rule, And All 10 Digits.

Codebook Rate (R)	Number Of Imposter Utterances	False Acceptances	Number Of Admissible Utterances	False Rejections
1	800	7	112	15
2	800	1	112	7
3	800	0	112	7
4	800	0	112	6

Table II. Speaker Verification Study: A Single Digit Codebook Per Speaker And R = 3.

Digit Spoken	Number Of Imposter Utterances	False Acceptances	Number Of Admissible Utterances	False Rejections	$\sqrt{FA*FR}$
ZERO	800	24 (3.0%)	112	10 (8.9%)	5.2
ONE	800	9 (1.1%)	112	19 (17.0%)	4.3
TWO	800	14 (1.8%)	112	10 (8.9%)	4.0
THREE	800	17 (2.1%)	112	17 (15.2%)	5.7
FOUR	800	12 (1.5%)	112	13 (11.6%)	4.2
FIVE	800	19 (2.4%)	112	26 (23.2%)	7.5
SIX	800	14 (1.8%)	112	17 (15.2%)	5.2
SEVEN	800	9 (1.1%)	112	16 (14.3%)	4.0
EIGHT	800	32 (4.0%)	112	15 (13.4%)	7.3
NINE	800	20 (2.5%)	112	20 (17.9%)	6.7

Table III. Speaker Verification Results: R = 3, Majority Rule, and All 10 Digits.

Speaker ID	Number Of Imposter Utterances	False Acceptances	Number Of Admissible Utterances	False Rejections	$\sqrt{FA*FR}$
TBS	222	0	10	0	0.0
WMF	222	0	10	0	0.0
RLD	222	1	10	0	0.0
GRD	222	1	10	1	2.1
KAB	222	5	10	2	6.7
MSW	222	0	10	0	0.0
REH	222	0	10	0	0.0
RGL	222	0	10	0	0.0
CJP	222	0	10	0	0.0
DFG	222	0	10	0	0.0
ALK	222	0	10	0	0.0
HNJ	222	0	10	0	0.0
GNL	222	0	10	1	0.0
JWS	222	0	10	0	0.0
SJN	222	0	10	1	0.0
SAS	222	0	10	1	0.0
Totals	3552	7 (0.2%)	160	6 (3.8%)	0.9

Table IV. Speaker Verification Results: Rate-3 Single Section Codebooks, Majority Rule, And 5 Digits.

Digit Subset	Number Of Imposter Utterances	False Acceptances	Number Of Admissible Utterances	False Rejections	$\sqrt{FA*FR}$
01247	3552	19 (0.5%)	160	6 (3.8%)	1.4
35689	3552	27 (0.8%)	160	4 (2.5%)	1.4
25678	3552	25 (0.7%)	160	8 (5.0%)	1.9

Table V. Speaker Verification Study: Multisection Codebooks, Majority Rule, All 10 Digits, 800 Imposter Utterances, and 112 Admissible Utterances.

Codebook	$n=12$		$n=8$		$n=4$		$n=2$		$n=1$	
Rate (R_S)	# FA	# FR	# FA	# FR	# FA	# FR	# FA	# FR	# FA	# FR
0	18	19	6	9	0	10	0	10	0	10
1	1	6	1	6	0	8	0	5	0	5
2	0	7	0	6	0	5	-	-	-	-

Table VI. Speaker Verification Study: A Multisection Digit Codebook Per Speaker, $R_S = 2$, and $n = 4$.

Digit Spoken	Number Of Imposter Utterances	False Acceptances	Number Of Admissible Utterances	False Rejections	$\sqrt{FA*FR}$
ZERO	800	19 (2.4%)	112	12 (10.7%)	5.1
ONE	800	11 (1.4%)	112	13 (11.6%)	4.0
TWO	800	8 (1.0%)	112	16 (14.3%)	3.8
THREE	800	21 (2.6%)	112	18 (16.1%)	6.5
FOUR	800	13 (1.6%)	112	13 (11.6%)	4.3
FIVE	800	16 (2.0%)	112	31 (27.7%)	7.4
SIX	800	9 (1.1%)	112	15 (13.4%)	3.8
SEVEN	800	6 (0.8%)	112	24 (21.4%)	4.1
EIGHT	800	33 (4.1%)	112	9 (8.0%)	5.7
NINE	800	16 (2.0%)	112	19 (17.0%)	5.8

Table VII. Speaker Verification Results: $R_S = 2$, $n = 4$, Majority Rule, and All 10 Digits.

Speaker ID	Number Of Imposter Utterances	False Acceptances	Number Of Admissible Utterances	False Rejections	$\sqrt{FA*FR}$
TBS	222	1	10	0	0.0
WMF	222	0	10	0	0.0
RLD	222	0	10	0	0.0
GRD	222	1	10	0	0.0
KAB	222	2	10	0	0.0
MSW	222	0	10	0	0.0
REH	222	0	10	0	0.0
RGL	222	0	10	0	0.0
CJP	222	2	10	0	0.0
DFG	222	0	10	0	0.0
ALK	222	3	10	0	0.0
HNJ	222	0	10	0	0.0
GNL	222	0	10	2	0.0
JWS	222	0	10	0	0.0
SJN	222	0	10	0	0.0
SAS	222	0	10	0	0.0
Totals	3552	9 (0.3%)	160	2 (1.3%)	0.6

Table VIII. Speaker Verification Results: Rate-2 Multisection Codebooks, Majority Rule, And 5 Digits.

Digit Subset	Number Of Imposter Utterances	False Acceptances	Number Of Admissible Utterances	False Rejections	$\sqrt{FA*FR}$
12467	3552	26 (0.7%)	160	1 (0.6%)	0.7
03589	3552	34 (1.0%)	160	5 (3.1%)	1.7
01234	3552	17 (0.5%)	160	3 (1.9%)	0.9

Table IX. Speaker Verification Study: Matrix Quantization Codebooks, Majority Rule, All 10 Digits, 800 Imposter Utterances, and 112 Admissible Utterances

Codebook	$K=4$		$K=8$		$K=12$		$K=24$	
Rate (R_M)	# FA	# FR	# FA	# FR	# FA	# FR	# FA	# FR
2	6	11	5	16	4	14	2	8
3	0	10	0	8	0	9	-	-
4	0	8	-	-	-	-	-	-

Table X. Speaker Verification Study: A Matrix Quantization Digit Codebook Per Speaker And $R_M=3$.

Digit Spoken	Number Of Imposter Utterances	False Acceptances	Number Of Admissible Utterances	False Rejections	$\sqrt{FA*FR}$
ZERO	800	24 (3.0%)	112	12 (10.7%)	5.7
ONE	800	6 (0.8%)	112	14 (12.5%)	3.2
TWO	800	12 (1.5%)	112	16 (14.3%)	4.6
THREE	800	29 (3.6%)	112	21 (18.8%)	8.2
FOUR	800	12 (1.5%)	112	13 (11.6%)	4.2
FIVE	800	45 (5.6%)	112	33 (29.5%)	12.9
SIX	800	11 (1.4%)	112	16 (14.3%)	4.5
SEVEN	800	11 (1.4%)	112	22 (19.6%)	5.2
EIGHT	800	36 (4.5%)	112	14 (12.5%)	7.5
NINE	800	17 (2.1%)	112	22 (19.6%)	6.4

Table XI. Speaker Verification Results: $R_M = 3$, $K = 8$, Majority Rule, and All 10 Digits.

Speaker ID	Number Of Imposter Utterances	False Acceptances	Number Of Admissible Utterances	False Rejections	$\sqrt{FA*FR}$
TBS	222	1	10	0	0.0
WMF	222	0	10	0	0.0
RLD	222	0	10	1	0.0
GRD	222	1	10	1	2.1
KAB	222	3	10	3	6.4
MSW	222	0	10	0	0.0
REH	222	0	10	0	0.0
RGL	222	2	10	0	0.0
CJP	222	0	10	0	0.0
DFG	222	0	10	0	0.0
ALK	222	0	10	0	0.0
HNJ	222	0	10	0	0.0
GNL	222	0	10	2	0.0
JWS	222	0	10	0	0.0
SJN	222	0	10	1	0.0
SAS	222	0	10	1	0.0
Totals	3552	7 (0.2%)	160	9 (5.6%)	1.1

Table XII. Speaker Verification Results: $R_M = 3$, Majority Rule, 5 Digits.

Digit Subset	Number Of Imposter Utterances	False Acceptances	Number Of Admissible Utterances	False Rejections	$\sqrt{FA*FR}$
12467	3552	17 (0.5%)	160	8 (5.0%)	1.5
03589	3552	44 (1.2%)	160	9 (5.6%)	2.6
01234	3552	15 (0.4%)	160	6 (3.8%)	1.3

References

1. Robert C. Lummis, "Speaker Verification by Computer Using Speech Intensity for Temporal Registration," *IEEE Transactions On Audio and Electroacoustics* **Vol. AU-21, No. 2**, pp. 80 - 89 (April 1973).

2. Sandra Pruzansky, "Pattern-Matching Procedure for Automatic Talker Recognition," *The Journal of the Acoustic Society of America* **Vol. 35, No. 3**, pp. 354 - 358 (March 1963).

3. B. S. Atal, "Effectiveness of Linear Predictive Characteristics of the Speech Wave for Automatic Speaker Identification and Verification," *J. Acoust. Soc. Am.* **Vol. 55**, pp. 1034-1312 (1974).

4. William S. Mohn, Jr., "Two Statistical Feature Evaluation Techniques Applied to Speaker Identification," *IEEE Transactions on Computers* **Vol. C-20, No. 9**, pp. 979 - 987 (September 1971).

5. P. D. Bricker *et al*, "Statistical Techniques for Talker Identification," *The Bell System Technical Journal* **Vol. 50, No. 4**, pp. 1427 - 1454 (April, 1971).

6. H. L. Van Trees, *Detection, Estimation and Modulation Theory - Part 1,* Wiley, New York, N.Y. (1968).

7. Robert M. Gray, "Vector Quantization," *ASSP Magazine*, pp. 4-29 (April 1984).

8. A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-28**, pp. 562-574 (Oct. 1980).

9. J. E. Shore and D. K. Burton, "Discrete utterance speech recognition without time alignment," *IEEE Trans. Inform. Theory* **IT-29**, pp. 473-491 (July, 1983).

10. N. Sugamura, K. Shikano, and S. Furiu, "Isolated Word Recognition Using Phoneme-Like Templates," pp. 723-726 in *Proceedings of ICASSP 1983, IEEE International Conference on Acoustics, Speech, and Signal Processing,* Boston, Mass. (April, 1983).

11. L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, "On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition," *The Bell Systems Technical Journal* **Vol. 62, No. 4**, pp. 1075-1105 (April, 1983).

12. Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.* **COM-28**, pp. 84-95 (Jan. 1980).

13. S. Kullback, *Information Theory and Statistics,* Dover, New York (1968). (Wiley, New York, 1959)

14. J. E. Shore and R. M. Gray, "Minimum-cross-entropy pattern classification and cluster analysis," *IEEE Trans. Patt. Anal. and Machine Intell.* **PAMI-4**, pp. 11-17 (Jan. 1982).

15. J. T. Buck, D. K. Burton, and J. E. Shore, "Text-Dependent Speaker Recognition Using Vector Quantization," pp. 11.5.1 - 11.5.4 in *Proceedings of 1985 ICASSP, IEEE International Conference on Acoustics, Speech, and Signal Processing,* Tampa, FL (March, 1985).

16. D. K. Burton, J. E. Shore, and Joseph T. Buck, "Isolated-Word Speech Recognition Using Multi-Section Vector Quantization Code Books," *IEEE Trans. Acoust., Speech, Signal Processing* (August, 1985). to appear

17. D. K. Burton, "Applying Matrix Quantization To Isolated word Recognition," pp. 1.8.1 - 1.8.4 in *Proceedings of ICASSP 1985, IEEE International Conference on Acoustics, Speech,*

and Signal Processing, Tampa, FL (March, 1985).

18. F. Soong *et al*, "A Vector Quantization Approach To Speaker Recognition," pp. 11.7-1 - 11.7-4 in *Proceedings of 1985 ICASSP, IEEE International Conference on Acoustics, Speech, and Signal Processing*, Tampa, Florida (March, 1985).

19. C. Tsao and R. M. Gray, "Matrix quantizer design for LPC speech using the generalized Lloyd algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing* (to appear 1986).

20. R. M. Gray, A. Buzo, A. H. Gray, Jr., and Y. Matsuyama, "Distortion measures for speech processing," *IEEE Trans. Acoust., Speech, Signal Processing* **ASSP-28**, pp. 367-376 (August 1980).

21. *Analog-to-Digital Conversion Of Voice By 2,400 Bit/Second Linear Predictive Coding*, General Services Administration (November 28, 1984). Federal Standard (FED-STD) 1015

22. Joseph T. Buck, "Vector quantization code book distortions as features for maximum likelihood classification of isolated words," pp. 9.3.1-9.3.5 in *Proceedings of 1984 IEEE Global Telecommunications Conference (GLOBECOM)*, Atlanta, GA (Nov. 1984).

23. G. R. Doddington and T. B. Schalk, "Speech recognition: turning theory to practice," *IEEE Spectrum* **Vol 18, No. 9**, pp. 26-32 (Sept. 1981).

24. R. Gary Leonard, "A Database for Speaker-Independent Digit Recognition," pp. 42.11.1-42.11.4 in *Proceedings of 1984 ICASSP, IEEE International Conference on Acoustics, Speech, and Signal Processing*, San Diego, California (March, 1984).

25. L. R. Rabiner and M. Sambur, "An algorithm for determining the endpoints of isolated utterances," *Bell Syst. Tech. J.* **54**, pp. 297-315 (Feb., 1975).

26. L. Lamel, *et al.*, "An improved endpoint detector for isolated word recognition," *IEEE Trans. Acoustics, Speech, & Signal Processing* **ASSP-29**, pp. 777-785 (Aug., 1981).

27. G. R. Doddington, "Voice Authentication Gets the Go-Ahead for Security Systems," *Speech Technology* **2**, pp. 14-23 (September/October 1983).

28. B. S. Atal, "Automatic Speaker Recognition Based On Pitch Contours," *J. Acoust. Soc. Am.* **Vol. 52**, pp. 1687-1697 (1972).